

An Enhanced Interpolated-Modulated Sliding DFT for High Reporting Rate PMUs

Paolo Romano, Mario Paolone

Distributed Electrical System laboratory (DESL)

Swiss Federal Institute of Technology of Lausanne (EPFL)

Lausanne, Switzerland

paolo.romano@epfl.ch, mario.paolone@epfl.ch

Abstract—The field of application of Phasor Measurement Units (PMUs) might be limited by the PMU measurements reporting latencies and achievable reporting rates, particularly with respect to power system protection applications that typically require very low latencies. A way to speed-up synchrophasor estimation algorithms based on the use of the Discrete Fourier Transform (DFT) refers to the usage of stable and accurate recursive processes for the DFT estimation. In this respect, this paper presents a synchrophasor estimation algorithm, called Interpolated-Modulated Sliding DFT (IpMSDFT), characterized by high accuracies and reduced latencies, enabling reporting rates up to thousands of synchrophasor per second. It is composed by two stages: (i) a guaranteed-stable technique for sample-by-sample DFT computation; (ii) an enhanced version of the classical IpDFT algorithm for synchrophasor estimation. The algorithm is analytically formulated and its digital design tailored to allow a feasible deployment on an FPGA-based PMU. The IpMSDFT-based PMU is finally validated with respect to the numerical stability of the proposed solution, its reporting latencies and the achievable reporting rates.

I. INTRODUCTION

The Phasor Measurement Unit (PMU) technology is constantly evolving and already composes the backbone of the most advanced metering systems in power networks.

One of the applications that might take major advantages from the availability of PMU data is power system protection. In this respect the literature already contains several contributions related to real-time fault identification and location processes using synchrophasors (e.g., [1]–[4]). The majority of these methods assumes a centralized approach where PMUs stream data to a Phasor Data Concentrator (PDC), that processes them in order to identify and locate the fault with the minimum possible latency.

If the centralized approach is adopted, in order to keep, at least, the same latencies characterizing existing relay-based protection schemes, the involved PMU technology has to satisfy additional requirements compared to those introduced by the IEEE Std. C37.118.1-2011 [5] and its latest amendment C37.118.1a-2014 [6]. In particular, it needs to estimate and report real-time synchrophasor-data with sufficient time-determinism, lower time-latencies and higher refresh rates without sacrificing their accuracy that has, at least, to keep the same levels dictated by [5] and [6]. Since the time spent in estimating the synchrophasor usually represents one of the main burden within the whole PMU-based measurement

chain, it is definitely necessary to develop faster synchrophasor estimation algorithms.

Most of the synchrophasor estimation algorithms that can be found in literature are based on the Discrete Fourier Transform (DFT). The accuracy of this category of methods has been widely discussed in the literature and several variations of this approach have been proposed, each one with its advantages and disadvantages. A specific technique combining the use of time-windows with the well-known Interpolated-Discrete Fourier Transform (IpDFT) technique has been first proposed in [7], [8] and further developed in [9]–[12]. This technique has demonstrated to be characterized by a relatively low computational complexity, and, above all, to be capable of achieving an optimal trade-off between the estimation accuracy and response-time.

Based on the adopted method to update the DFT estimation, IpDFT algorithms can be separated into two main categories: recursive and non-recursive algorithms.

Within the group of non-recursive algorithms the well known Fast Fourier Transform (FFT) algorithm [13] is widely used. Typically, this implementation is adopted to perform harmonic analysis over an extended portion of the spectrum even though its deployment on embedded system is usually onerous. When, on the other hand, only a subset of the overall DFT spectrum is used to estimate the synchrophasor (see for instance [8]), the classic DFT theory turns out to be very effective. In both cases, the measurement reporting latencies are proportional to the adopted window length. As a consequence, the algorithm throughput can only be improved at the cost of reducing the window length, i.e., deteriorating the PMU accuracy levels [11].

In order to increase the throughput without decreasing the precision of the adopted IpDFT synchrophasor estimation algorithm, DFT can be calculated via recursive algorithms that are usually characterized by a lower number of operations to update the values of a single DFT bin (e.g., [14]). Despite this evident advantage with respect to the class of non-recursive DFT algorithms, the two categories do not generally have identical performances. In particular, the majority of the recursive algorithms suffers of errors due to either the approximations made to perform the recursive update, or the accumulation of the quantization error produced by the finite word-length of computers [15].

A very effective method for sample-by-sample DFT bins computation, is represented by the so-called Sliding-DFT (SDFT) technique presented in [16]. This reference demonstrates the efficiency of this method in comparison with the popular Goertzel algorithm and its computational advantages over the more traditional DFT and FFT, but also its drawbacks. In particular, the approach proposed in [16] is only marginally stable. In case the filter coefficient numerical rounding error is not severe, the SDFT is bounded-input, bounded-output stable. Otherwise, the algorithm suffers from accumulated errors due to numerical rounding and is, consequently, potentially unstable. Whereas common approaches found in literature [16], [17] face this problem compromising results accuracy for guaranteed stability, the method proposed in [18] and called Modulated Sliding DFT (MSDFT) is guaranteed stable without sacrificing accuracy.

In the framework of the above-listed literature, the paper focuses on the challenge of defining a high-speed synchrophasor estimation algorithm characterized at the same time by high accuracy, high throughputs and low-computational complexity. This algorithm, hereafter called Interpolated-Modulated Sliding DFT (IpMSDFT), besides maintaining the accuracy levels first shown by the authors in [19], aims at achieving reporting rates in the range of thousands of synchrophasors per seconds. In this respect, the aim of the paper is threefold: (i) extend the MSDFT approach described in [18] to the case of synchrophasor estimation; (ii) couple this approach with the enhanced interpolated-DFT method proposed by the Authors in [19]; (iii) validate the proposed approach on a Field Programmable Gate Array (FPGA) based PMU prototype.

The structure of the paper is the following: Section II and III present the analytical formulation and a possible digital design of the IpMSDFT algorithm respectively; Section IV experimentally validates the IpMSDFT algorithm when deployed into a PMU prototype. Finally Section V concludes the paper with the remarks and conclusions.

II. SYNCHROPHASOR ESTIMATION ALGORITHM FORMULATION

Any IpDFT algorithm is characterized, at least, by 3 sequential stages represented by: (i) the signal windowing, (ii) the DFT calculation and (iii) the DFT interpolation. Since signal windowing can be applied either in time or frequency domain [20], stages (i) and (ii) can be equivalently exchanged. In this respect, the following Section is arranged in 2 parts: the first part presents the MSDFT algorithm and adapts it to the case of synchrophasor estimation; the second one couples the MSDFT method with the enhanced IpDFT approach presented by the Authors in [19].

A. MSDFT for Synchrophasor Estimation

A power system quantity (branch current or node voltage) can be modeled as a signal characterized by a main tone fluctuating around the rated frequency of the system f_0 (i.e., 50 or 60 Hz). The signal is sampled by the PMU each $T_s = 1/F_s$ (being F_s the PMU sampling rate), and collected over the

sliding window of length M sufficiently short so that the signal can be assumed stationary within it:

$$x(m) = A \cos(2\pi f m T_s + \varphi), \quad m \in [0, M-1] \quad (1)$$

where A , f and φ are the amplitude, frequency and phase of the main tone of the spectrum that are supposed to be constant over the observation interval of length M .

At every time-step n , the DFT can be potentially updated based on the most recent set of samples $\{x(n-M+1), x(n-M+2), \dots, x(n)\}$:

$$X_k(n) = \sum_{m=0}^{M-1} x(q+m) \cdot W_M^{-km} \quad (2)$$

being k the DFT-bin index, $q = n - M + 1$ and $W_M^{-km} = e^{-j2\pi km/M}$ the DFT complex twiddle factor.

As demonstrated in [16], the so-called Sliding-DFT (SDFT) formula can be derived from (2) as:

$$\begin{aligned} X_k(n) &= \sum_{m=0}^{M-1} x(q+m) \cdot W_M^{-km} \\ &= \sum_{m=0}^{M-1} x(q+m-1) \cdot W_M^{-k(m-1)} + \\ &\quad - x(q-1) \cdot W_M^{-k} + x(q+M-1) \cdot W_M^{-k(M-1)} \\ &= W_M^{-k} \cdot \sum_{m=0}^{M-1} x(q+m-1) \cdot W_M^{-k(m-1)} + \\ &\quad - x(q-1) \cdot W_M^{-k} + x(q+M-1) \cdot W_M^{-k(M-1)} \\ &= W_M^{-k} \cdot (X_k(n-1) - x(q-1) + x(q+M-1)) \\ &= W_M^{-k} \cdot (X_k(n-1) - x(n-M) + x(n)) \end{aligned} \quad (3)$$

The recursive DFT calculation described by (3) is potentially unstable and might suffer from accumulated errors due to numerical rounding. However, as shown in [18], it is easy to observe that, when $k = 0$, the recursive formula for the computation of X_k expressed by Equation (3) does not involve the complex twiddle factor and is, therefore, by definition stable:

$$X_0(n) = X_0(n-1) - x(n-M) + x(n) \quad (4)$$

By taking advantage of this property, and making use of the so-called Fourier modulation property [21], the generic k -th DFT-bin can be shifted to the position $k = 0$ multiplying the input signal $x(n)$ by the complex twiddle factor W_M^{-km} :

$$\begin{aligned} X_0(n) &= X_0(n-1) + \\ &\quad - x(n-M) \cdot W_M^{-k(m-M)} + x(n) \cdot W_M^{-km} \\ &= X_0(n-1) + \\ &\quad + W_M^{-km} \cdot (-x(n-M) + x(n)) \end{aligned} \quad (5)$$

where equation (5) is obtained thanks to the intrinsic periodicity of the modulating sequence W_M^{-km} .

The twiddle factor modulation only introduces a phase shift that is changing with index m : it is equal to zero for $m = 0$,

it increases by the $\angle W_M^{-k}$ factor at each iteration and is periodically reset to 0 every M samples. In view of this, the k -th bin of the DFT can be derived from equation (5) as:

$$X_k(n) = W_M^{k(m+1)} \cdot X_0(n) \quad (6)$$

where $W_M^{-k(m+1)}$ compensate for the phase-shift due to the modulating sequence.

Equations (5) and (6) define the MSDFT method for the update of the value of a single bin of the entire DFT spectrum.

B. Enhanced-IpMSDFT Algorithm

During power-system dynamics, the spectral leakage effects decrease the accuracy levels of the majority of DFT-based synchrophasor estimation algorithms well above the IEEE Standard C37.118.1 limits. IpDFT algorithms try to reduce this bias by sequentially applying specific windowing functions and DFT interpolation schemes but still suffer of errors when the frequency of the signal drifts from the rated one. Indeed, every IpDFT algorithm is based on the assumption that $F_s \gg f_0$. As a consequence the positive and negative image of the main tone are typically very close in the DFT spectrum and the tails of their envelop might eventually corrupt the neighboring image. In this respect, the enhanced-IpDFT (e-IpDFT) scheme presented in [19] extends the classical IpDFT approach by compensating the spectral interference produced by the negative image of the spectrum. It can be described in terms of the following simplified procedure:

```

1: procedure E-IPDFT( $x(m)$ ,  $m \in [0, M - 1]$ )
2:   apply Hanning window
3:   compute DFT
4:   estimate signal parameters via 2-points IpDFT
5:   for  $i \leftarrow 1, P$  do
6:     estimate spectral interference
7:     compensate for spectral interference
8:     estimate signal parameters via 2-points IpDFT
9:   end for
10:  return signal parameters
11: end procedure

```

In [19] it has been shown that, by adopting a time window T containing 3 exact periods of a signal at the rated power system frequency (i.e., 50 or 60 Hz) and a sampling rate F_s of some tens of kHz, the IpDFT algorithms performs well under most of the conditions dictated by [5]. In particular the developed e-IpDFT-based PMU prototype is capable of passing every test defined by [5] for both PMU classes P and M except the out-of-band ones.

The measurement reporting latencies and achievable reporting rates of the e-IpDFT algorithm are mainly limited by the time needed to compute the relevant portion of the DFT spectrum. In particular, as shown in [19], the e-IpDFT algorithm only needs to compute the 3 DFT bins associated to indices $k_{max} + \{-1, 0, 1\}$, where $k_{max} \in \mathbb{N}$ is the index of the DFT maximum that is fixed for typical PMU operating

conditions and equal to $k_{max} = \lceil f_0 M / F_s \rceil$, being $\lceil \cdot \rceil$ the nearest integer function.

In this respect the MSDFT seems to fit well in the e-IpDFT scheme that might benefit from its fast refresh rates without disrupting the previously defined procedure. The only operation that needs careful consideration is the signal windowing (i.e. step 2 of the e-IpDFT procedure), since its application in time-domain would compromise the whole MSDFT formulation. As a consequence it needs to be moved after the MSDFT update, and replaced by a frequency-domain convolution that results into a linear combination of adjacent $X_k(n)$ values. In the case of the Hanning window, the windowed k -th bin can be computed as:

$$X_k(n) = -0.25 \cdot X_{k-1}(n) + 0.5 \cdot X_k(n) - 0.25 \cdot X_{k+1}(n) \quad (7)$$

From Equation (7), it is clear that, in order to compute 3 windowed DFT bins, we need to compute 5 MSDFT bins, namely those associated to indices $k_{max} + \{-2, -1, 0, 1, 2\}$. Therefore, we can expect that the MSDFT would not modify the precision of the e-IpDFT algorithm but only improve its measurement reporting latencies and achievable reporting rates.

III. E-IPMSDFT DIGITAL DESIGN

The MSDFT assumes that for each new sample, every DFT bin is updated in order not to compromise the next DFT estimations. This computation must be performed before the acquisition of the next sample, over the whole set of PMU input channels, in order to correctly estimate the corresponding synchrophasors. Since the PMU technology has to provide estimations with a very high degree of confidence, the platform that will host the IpMSDFT algorithm need to guarantee a certain level of determinism.

In this respect, the FPGA technology currently represents the best solution to host such a sample-by sample DFT calculation for several reasons: (i) the high-speed clock characterizing such a platform and allowing millions of operation per second; (ii) its intrinsic parallelism and determinism that guarantee constant latencies with any input condition; (iii) the increased size and complexity of the logical blocks composing the FPGA that nowadays embeds a significant set of DSP block.

In what follows we will present the deployment of the IpMSDFT algorithm in a PMU prototype based on the Xilinx Artix-7 FPGA embedded into the Xilinx Zynq 7020 System on a Chip (SoC). In particular, in the coming tests, the window length and the sampling rate are set to 60 ms (3-periods of a signal at a rated frequency of 50 Hz) and 10 kHz respectively (i.e., $M = 600$). The analysis will focus on 2 major items: (i) the FPGA resources allocation and (ii) the UTC time synchronization of the IpMSDFT algorithm.

A. FPGA resources allocation

A minimum set of FPGA logical blocks needed to implement the MSDFT algorithm for the update of a single DFT bin is shown in Fig. 1 together with the relevant interconnections.

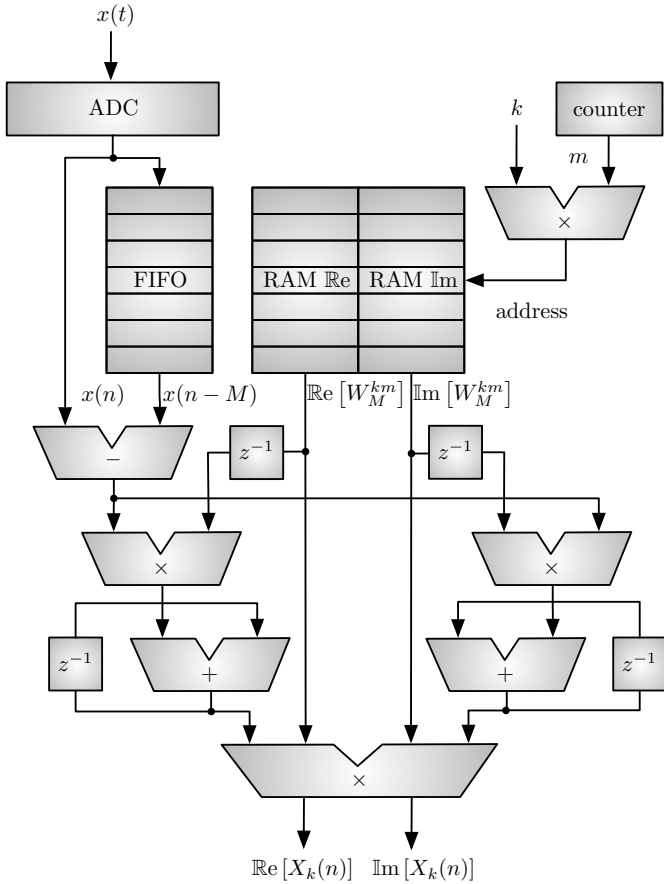


Fig. 1. Logical blocks needed to compute a single DFT bin by means of the MSDFT algorithm.

The analog signal $x(t)$ is digitally converted by an A/D converter to provide the digital sequence of samples $x(n)$. As shown in equation (5), the MSDFT has to keep memory of the last M samples in order to compute the difference $x(n) - x(n - M)$. This can be done by means of a FIFO (First-In-First-Out) memory. In parallel, every time a new sample comes, a modular counter increments the value of the index $m \in [0, M - 1]$. This, together with the value of the DFT bin to be computed, returns the address of the memory location of the a pre-allocated RAM that contains the instantaneous values of the twiddle factor W_M^{km} , expressed in terms of its real and imaginary part. Rather than recomputing the twiddle factor at every iteration, such a solution has two main advantages: (i) it saves execution time by sacrificing the occupation of the available blocks of RAM inside the FPGA; (ii) the twiddle factor computation does not suffer of any accumulated errors. Once the above listed operations completed, the k -th DFT bin $X_k(n)$ can be updated based on equations (5), and (6) separately for real and imaginary part.

Based on the block scheme of Figure 1, the MSDFT method for the estimation of a single DFT bin was deployed in the FPGA. As it can be noticed from the compilation results shown in the first row of Table I, only with the allocated resources for the update of a single DFT bin via MSDFT, the number of used DSP blocks raises up to 41. Since we need to compute 5 bins of the DFT spectrum, by replicating 5 times the previously

TABLE I
FPGA OCCUPATION OF THE E-IPMSDFT COMPONENTS

Logical blocks	Flip-Flops	LUTs	DSP ¹	RAM ²	latency [μ s]
MSDFT (1-bin)	16,200	17,090	41	7	0.5
MSDFT (5-bins)	17,966	19,041	41	7	2.5
e-IpDFT (SE only)	15,236	16991	81	0	19.2
Total available	106,400	53,200	220	140	-

¹ One DSP contains a 18x25 multiplier.

² One block RAMs can store up to 32 kbits of data.

designed block we would get very close to the FPGA physical limits and would not be able to add on top of that the e-IpDFT algorithm. Hence the calculation of the 5 DFT bins needs to be serialized by sharing the portion of the FPGA dedicated to the MSDFT computation with every DFT bin and sacrificing the algorithm latencies for the FPGA area requirements. Since the latency of the proposed MSDFT implementation is very small (0.5 μ s) this choice would not add much in term of measurement reporting latency.

The compilation results of a 1-channel MSDFT that computes the DFT bins associated to the DFT indices $k_{max} + \{-2, -1, 0, 1, 2\}$ based on the above explained logic are shown in the second row of Table I. As expected, the usage of FPGA logic blocks does not significantly change compared to the single DFT update, whereas the latency increases by exactly a factor of 5. As a consequence the proposed solution allows to estimate 5 DFT bins of a maximum possible number of 37 channels at a sampling rate of 10 kHz. To be noticed that the extension to more than 1 input channel will only increase the amount of used RAM needed to keep memory of the latest M samples for each input channel. On the other hand, the amount of memory needed to store the pre-computed twiddle factor will not change, as well as the amount of used DSP blocks.

The MSDFT calculation and the e-IpDFT synchrophasor estimation algorithm can be decoupled and run in two separated processes. The one that estimates the synchrophasor will be activated based on the reporting rate settings and after the MSDFT has been updated. The compilation results of the portion of the e-IpMSDFT algorithm that estimates the synchrophasor based on the most recent set of DFT bins are also shown in Table I for a PMU equipped with a single input channel. The characteristic latency of this portion of the FPGA is of only 19.2 μ s. Therefore the PMU can estimates synchrophasor at a maximum reporting rate equal, for the current setup, at the adopted sampling rate F_s .

B. UTC-synchronization of the e-IpMSDFT estimations

The MSDFT needs to run continuously to correctly update the values of a set of DFT bins, and therefore potentially obtain a new estimation every time a new raw sample is acquired. In order to align the PMU estimations with the UTC time, some specific considerations must be made.

As shown in Figure 2, the synchronization of the PMU estimations to UTC-time can be performed by internally synthesizing a square waveform, hereafter called subPPS,

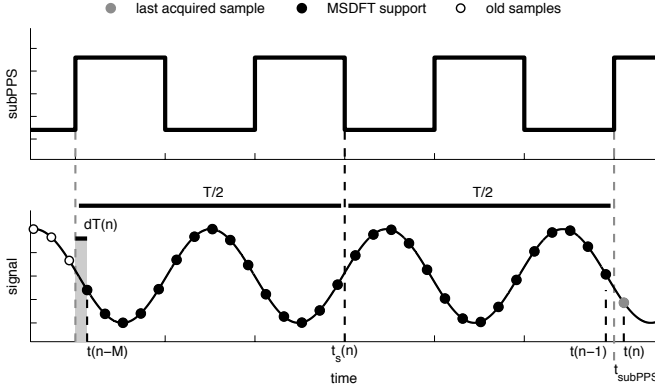


Fig. 2. An intuitive representation of the approach used to synchronize the e-IpMSDFT estimations to the UTC-time.

aligned to the UTC-PPS (Pulse-Per-Second) but shifted back by half of the window length T and characterized by a frequency equal to the PMU reporting rate. The rising edge of the subPPS waveform triggers the e-IpDFT synchrophasor estimation algorithm, that uses the second last MSDFT update (i.e. the one based on the set of samples centered around the reporting time) to provide the most recent PMU estimation.

In practice, if we assume that the latest acquired sample $x(n)$ is characterized by a UTC time-stamp value $t(n)$, based on the above-described procedure, the synchrophasor's time-stamp $t_s(n)$ is the time corresponding to the middle of the time window of length T that contains the set of samples $\{x(n-M), \dots, x(n-1)\}$ that can be computed as:

$$t_s(n) = t(n-1) + T_s - T/2 \quad (8)$$

where T_s is the PMU sampling time.

In the case of the proposed e-IpMSDFT algorithm, the PMU sampling process can be either synchronized to an absolute and stable time reference like the one provided by the Global Positioning System or free-running. In the first case, the above explained logic stands well alone: since the sampling process is synchronized to the subPPS rising edges, the time-stamp value computed using equation (8) corresponds to the PMU reporting time and the e-IpMSDFT estimations are automatically aligned to the UTC-time.

On the other hand, if the sampling process is free-running, based on the environmental conditions, it will drift and the 1st sample of the window will be rarely aligned to the subPPS (see Figure 2). As demonstrated in [19], in order to correct the effects of the free-running sampling process, 2 main countermeasure must be taken:

- the time-stamp value, together with the estimated phase, must be compensated for the following time difference:

$$dT(n) = t(n-M) - (t_{subPPS} - T) \quad (9)$$

otherwise their accuracy will be proportional to the sampling time T_s (see Figure 2);

- in order to improve the frequency (and therefore the phase) estimations, the window length needs to be measured in real time and, as the sampling process drifts, opportunely compensated in the e-IpDFT scheme.

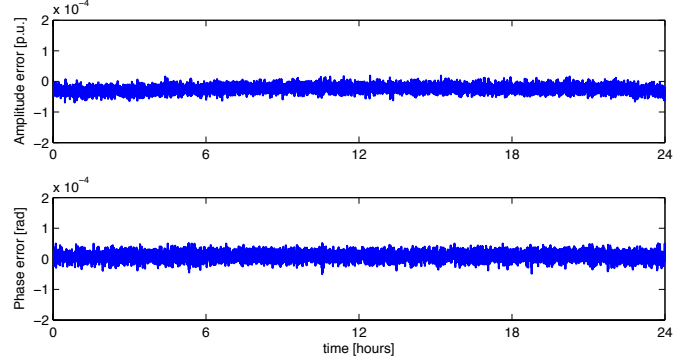


Fig. 3. e-IpMSDFT numerical stability over 24 hours.

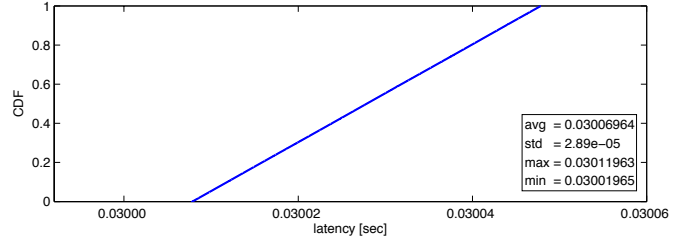


Fig. 4. Cumulative distribution function of the measurement reporting latencies for the proposed e-IpMSDFT algorithm.

IV. E-IPMSDFT EXPERIMENTAL VALIDATION

In this Section the e-IpMSDFT algorithm and the relevant implementation are experimentally validated with respect to: (i) the e-IpMSDFT stability over time, (ii) its measurements reporting latencies and (iii) the achievable reporting rates.

The validation has been carried out using a National Instruments PXI system opportunely synchronized to GPS as a reference waveforms generator (see [19] for further details).

A. Numerical stability

During this test the e-IpMSDFT was left running for 24 hours with a reporting rate of 1 frame per second. The PXI-based reference signal generator was set to synthesize a GPS-aligned 50 Hz waveform. During the test the estimated amplitude and phase values were collected and compared with the reference phasors. Figure 3 shows the errors over the 24 hour test. As it can be noticed the estimations do not show any instability and the proposed MSDFT design can be considered correct.

B. e-IpMSDFT measurement reporting latencies

The measurement reporting latency is defined as the time-delay from when an event occurs in the power system to the time that it is reported in the data and can be computed as the time interval between the data time-stamp and the time when the data becomes available to the PMU outputs (see [5]).

Figure 4 shows the cumulative distribution function, computed over 10,000 consecutive estimations, of the measured latencies of a IpMSDFT-based PMU that was running at a reporting rate of 50 estimations per seconds (i.e., the test covers a time interval of more than 3 minutes). As it can be noticed the distribution of the measurement reporting latencies

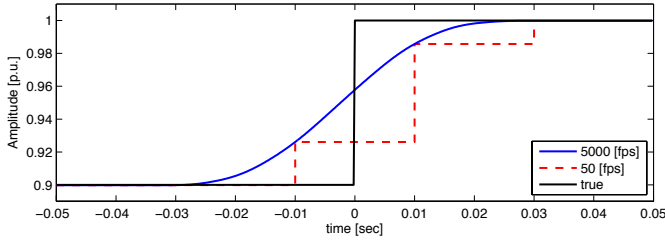


Fig. 5. Estimated amplitude profiles during an amplitude step by PMUs characterized by reporting rates of 50 and 5000 [fps] respectively.

is uniform with an average value of 0.030069 seconds, in agreement with the expected one:

$$T_m = T/2 + T_{\text{proc}} \quad (10)$$

being T the windows length that, as a recall, is set to 60 ms and T_{proc} the processing time. Note that T_{proc} is slightly larger than the sum of the synchrophasor computation latencies shown in Table I, in view of the communication delays between the MSDFT and synchrophasor estimation processes.

C. e-IpMSDFT reporting rates

As demonstrated in Section III-A the e-IpMSDFT-based PMU prototype built on the proposed design, can achieve reporting rates up to the PMU sampling rate F_s . In this test the advantages brought by this outstanding characteristic of the proposed PMU prototype are verified during a step change in the amplitude of the input signal. In particular, Figure 5 shows the estimated amplitude during such a sudden change by two different PMUs: the first one (grey-continuous line) runs the e-IpDFT algorithm presented in [19] that is characterized by a maximum reporting rate of 50 frames-per-second; the second one (gray-dashed line) runs the proposed e-IpMSDFT algorithm at a reporting rate of 5000 estimated synchrophasors per second. As it can be noticed the e-IpMSDFT algorithm offers a time-resolution proportional to the reporting rate. Obviously, the MSDFT does not affect the PMU response times during the step tests that will maintain the same values demonstrated in [19].

V. CONCLUSIONS

The paper has presented an efficient method based on MSDFT capable of accelerating the reporting rates of any DFT-based synchrophasor estimation algorithm up to the limits of the PMU sampling rate. In particular, the MSDFT method was combined with the e-IpDFT synchrophasor estimation algorithm, previously proposed by the authors in [19]. The novel algorithm has been formulated and its deployment into an FPGA-based PMU prototype illustrated. The validation of the proposed method has demonstrated the reduced reporting latencies and extremely high reporting rates that the e-IpMSDFT algorithm can achieve.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7-ICT-2011-8 under grant agreement n° 318708 (C-DAX).

REFERENCES

- [1] C.-S. Chen, C.-W. Liu, and J.-A. Jiang, "A new adaptive PMU based protection scheme for transposed/untransposed parallel transmission lines," *Power Delivery, IEEE Transactions on*, vol. 17, no. 2, pp. 395–404, 2002.
- [2] C.-S. Yu, C.-W. Liu, S.-L. Yu, and J.-A. Jiang, "A new PMU-based fault location algorithm for series compensated lines," *Power Delivery, IEEE Transactions on*, vol. 17, no. 1, pp. 33–46, 2002.
- [3] J.-A. Jiang, J.-Z. Yang, Y.-H. Lin, C.-W. Liu, and J.-C. Ma, "An adaptive PMU based fault detection/location technique for transmission lines. i. theory and algorithms," *Power Delivery, IEEE Transactions on*, vol. 15, no. 2, pp. 486–493, 2000.
- [4] J.-A. Jiang, Y.-H. Lin, J.-Z. Yang, T.-M. Too, and C.-W. Liu, "An adaptive PMU based fault detection/location technique for transmission lines. ii. PMU implementation and performance evaluation," *Power Delivery, IEEE Transactions on*, vol. 15, no. 4, pp. 1136–1146, 2000.
- [5] "IEEE standard for synchrophasor measurements for power systems," *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–61, 2011.
- [6] "IEEE standard for synchrophasor measurements for power systems – Amendment 1: Modification of selected performance requirements," *IEEE Std C37.118.1a-2014 (Amendment to IEEE Std C37.118.1-2011)*, pp. 1–25, April 2014.
- [7] V. K. Jain, W. L. Collins, and D. C. Davis, "High-accuracy analog measurements via interpolated FFT," *Instrumentation and Measurement, IEEE Transactions on*, vol. 28, no. 2, pp. 113–122, 1979.
- [8] T. Grandke, "Interpolation algorithms for Discrete Fourier Transforms of weighted signals," *Instrumentation and Measurement, IEEE Transactions on*, vol. 32, no. 2, pp. 350–355, 1983.
- [9] D. Belega and D. Dallet, "Multifrequency signal analysis by interpolated DFT method with maximum sidelobe decay windows," *Instrumentation and Measurement, IEEE Transactions on*, vol. 42, no. 3, pp. 420–426, 2009.
- [10] M. Paolone, A. Borghetti, and C. A. Nucci, "A synchrophasor estimation algorithm for the monitoring of active distribution networks in steady state and transient conditions," in *Proc. of the 17th Power Systems Computation Conference (PSCC 2011)*, Stockholm, Sweden, Aug, 2011.
- [11] D. Belega and D. Petri, "Accuracy analysis of the multicycle synchrophasor estimator provided by the interpolated DFT algorithm," *Instrumentation and Measurement, IEEE Transactions on*, vol. 62, no. 5, pp. 942–953, 2013.
- [12] D. Belega, D. Macii, and D. Petri, "Fast synchrophasor estimation by means of frequency-domain and time-domain algorithms," *Instrumentation and Measurement, IEEE Transactions on*, vol. PP, no. 99, 2013.
- [13] J. W. Cooley, P. A. Lewis, and P. D. Welch, "The Fast Fourier Transform and its applications," *Education, IEEE Transactions on*, vol. 12, no. 1, pp. 27–34, 1969.
- [14] A. G. Phadke and J. S. Thorp, *Synchronized phasor measurements and their applications*. Springer, 2008.
- [15] H. A. Darwish and M. Fikri, "Practical considerations for recursive DFT implementation in numerical relays," *Power Delivery, IEEE Transactions on*, vol. 22, no. 1, pp. 42–49, 2007.
- [16] E. Jacobsen and R. Lyons, "The sliding DFT," *Signal Processing Magazine, IEEE*, vol. 20, no. 2, pp. 74–80, 2003.
- [17] S. Douglas and J. Soh, "A numerically-stable sliding-window estimator and its application to adaptive filters," in *Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, vol. 1. IEEE, 1997, pp. 111–115.
- [18] K. Duda, "Accurate, guaranteed stable, sliding discrete fourier transform [DSP tips & tricks]," *Signal Processing Magazine, IEEE*, vol. 27, no. 6, pp. 124–127, 2010.
- [19] P. Romano and M. Paolone, "Enhanced interpolated-DFT for synchrophasor estimation in FPGAs: Theory, implementation, and validation of a PMU prototype," *Instrumentation and Measurement, IEEE Transactions on*, no. 99, pp. 1–13, 2014.
- [20] F. J. Harris, "On the use of windows for harmonic analysis with the Discrete Fourier Transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.
- [21] A. V. Oppenheim, R. W. Schaffer, J. R. Buck *et al.*, *Discrete-time signal processing*. Prentice-hall Englewood Cliffs, 1989, vol. 2.